2.10 The Network

The term "network" is used here to refer to the facilities used to connect the sender's and recipient's EDI systems. There are four basic kinds of network configurations used by trading partners:

1) Point-to-point. Two trading partners may communicate directly with one another through a dial-up common carrier network or a dedicated circuit. The sender's EDI system communicates directly with the recipient's EDI system. The "network" does not have a storage capability, and does not provide any message status information.

2) Use of a Single Value-Added Network (VAN). The trading partners use a common VAN to communicate. The typical VAN simplifies communications for a sending partner who has many receiving partners. The VAN accepts messages from the sender and passes them to the recipients; the sender does not have to contact each recipient separately. Each VAN user is said to have a "mailbox." When the VAN receives a message from a sender, it reads the address on the message "envelope" (header) to identify the recipient. The VAN then moves the messages from the sender's mailbox to the recipient's mailbox. Later the recipient's EDI system connects to the VAN, discovers the message and downloads it. This method of operation is often called "store-and-forward."

A VAN can report to a sender when it deposits a message in a recipient's mailbox, and when the recipient removes the message from his or her mailbox. This confirms to the sender that the recipient has the message, and helps to support the authentication of both the sender and recipient.

3) Use of Two VANs. If trading partners are users of different VANs, it may be possible to arrange for a VAN-to-VAN connection. Operation is the same as described above, except that the message must first move from the sender's VAN to the recipient's VAN. VANs typically maintain gateways to other VANs as a service to their subscribers. It is desirable for the sender's VAN to be able to report complete status information back to the sender about the delivery of a message to the recipient and the recipient's retrieval. If the two VANs cannot interchange complete information, then the sender's VAN may only be able to report to the sender that the message was passed to the recipient's VAN but not to the recipient. In this latter case, knowledge of timely delivery to the correct party is not assured to the sender. Use by the VANs of the X.400 communications protocol, or the X12 Committee's X12.56 Interconnect Mailbag Control Structures, may provide the necessary support to provide the needed information.

4) Dedicated Network. The dominant trading partner provides and operates the network that the subordinate trading partners use to send and receive EDI messages.

2.11 Potential Network Risks

The possibility of network hardware and software failures, misfeasance or malfeasance of network personnel, and actions by outsiders can result in risk. As noted below, some risks do not apply to all network types.

1) A message is delivered to the wrong recipient. This risk does not apply to messages from a subordinate partner to a dominant partner on a dedicated network, or on a dedicated point-to-point network.

2) Undetected corruption of a message occurs.

3) Failure of a message to reach the recipient is not detected. (Applies primarily to use of VANs.)

4) A VAN incorrectly reports to the sender the status of message pickup by the recipient. For example, the pickup occurred significantly later than reported, or was not reported when it occurred.

5) A message is delayed in transmission significantly longer than expected. What constitutes a significant delay will depend on the character of the message. If the network is a VAN, the usage agreement should specify the expected delivery time.

6) A message is intercepted and disclosed to others without authorization. This risk applies to all network types, but a wire-tap is not required on a VAN since messages typically are stored on back-up files, and VAN personnel routinely monitor traffic.

7) A message is intercepted and modified without authorization, and then transmitted on to the recipient.

2.12 The Recipient's EDI System

The recipient's EDI system performs functions similar to the sender's EDI system, but in the opposite sequence. The EDI system receives messages from the EDI network, translates the EDI transaction sets in the messages, e.g., one or more 850 Purchase Orders, into in-house formats, and passes them to the appropriate recipient's applications. The translations make use of maps to relate transaction set data elements to data fields of the transaction files passed to the applications.

The EDI system may also generate a 997 Functional Acknowledgment transaction set and transmit it to the sender. Note that the name of this transaction set is not fully descriptive. The sender can only conclude that the transaction set being acknowledged was received intact by the recipient, but not that it was accepted by a recipient application. For example, a functional acknowledgment of a purchase order transaction set does not constitute acceptance of the purchase order. The 997 Functional Acknowledgment is the EDI equivalent of a U.S. Postal Service return receipt.

2.13 Potential Risks of the Recipient's EDI System

The possibility of hardware and software failures of the recipient's EDI system, and misfeasance or malfeasance of EDI system personnel results in the following risks:

1) An EDI message is received from the network but not otherwise processed.

2) An EDI message is received from the network but no acknowledgment is sent as expected by the network or the sender's EDI system.

3) A transaction set is acknowledged as received, but is lost internally before it is passed to the correct recipient application system.

4) Incorrect translation of a transaction set is not detected. The wrong acknowledgment is sent.

2.14 The Recipient's Application

The recipient's application receives and acts on the translated transaction sets received from the recipient's EDI system. Functionally, this is the same as receiving the data from key-stroked, paper source documents.

If one of the transactions sets is an 850 Purchase Order, for example, the order entry application validates the transaction. If it is acceptable, the application generates an acknowledgment transaction, for example, an 855 Purchase Order Acknowledgment transaction set, and sends it back to the sender. In a fully re-engineered EC system, the order entry application might also transmit input data to the warehouse, inventory control, customer credit, accounts receivable, and shipping systems to fulfill the purchase order.

2.15 Potential Risks of the Recipient's Application

Hardware and software failures of the recipient's application result in the following risks:

1) An invalid or corrupted transaction is not detected.

2) Receipt of a valid transaction set is not acknowledged by the recipient as expected by the EDI system and/or the sender.

3) Receipt of a duplicate transaction set is not detected.

4) Invalid translation of a transaction set is not detected.

5) The application does not reconcile its table of transactions processed with the EDI system's table of transactions passed to the application.

2.16 Risks Not Specific to EC Systems

EC systems typically are connected to business data processing systems that relate to other activities. Examples of such data processing systems are those for finance, accounts payable and receivable, inventory and shipping. These traditional data processing systems are exposed to general risks that are not specific to EC systems, but that could affect them. Some of these risks are:

(1) Service interruptions to general data processing systems caused by risks such as hardware and software failures, fires, floods, earthquake, sabotage, etc.

(2) Application fraud due to staff personnel entering falsified transactions or data into general data processing systems or by modifying applications or operating system programs.

(3) Unauthorized disclosure of information, by means of reports or files generated or maintained by general data processing systems to which EC systems are connected.

These risks may already have been analyzed as a part of an existing risk management program. In any event, they should be included in the risk analysis of the EC system.

3. GOOD SECURITY PRACTICES

3.1 Summary

This chapter describes good security techniques that apply during the design, test, and operational phases of EC systems implementation, and it addresses the special requirements of EC systems. These techniques include subsystem-to-subsystem acknowledgments and other techniques, especially for the application, EDI, and network subsystems. In addition, access controls, electronic document management, audit trails, contingency plans, compliance audits, and system testing are discussed.

A security technique should not be adopted simply because it is described here. It should only be included in an EC system if it is expected to have a beneficial impact on the operating cost of the EC system. That is, it should be used if the expected reduction in losses will outweigh the cost to implement the security technique, or the security technique will address an unacceptably high single-occurrence loss.

3.2 Use of Acknowledgments

Use of acknowledgments is a good security practice; it is fundamental to secure EC because it addresses several important risks:

1) duplicated transaction sets generated in error by the sender's application or EDI system, the network, or the recipient's EDI system;

2) repudiated transaction sets;

3) lost transaction sets; and

4) invalid or corrupted transaction sets.

The most important risk addressed by an acknowledgment is the duplicate transaction set. A recipient cannot detect a transaction set that the sender's application has duplicated by mistake, since (as discussed in Section 3.3.1) the two transaction sets should have different sequence numbers. The expense of subsequent corrective action may be quite high. For example, a recipient may take a high-cost action, e.g., fabricate custom-designed parts, in response to an undetected duplicate purchase order. However, a detailed acknowledgment of the inadvertently duplicated transactions should enable the sender to detect the duplication and take prompt corrective action.

Every EC message should be acknowledged with a message from the recipient's application sent back to the sender's application, within a stipulated time defined in the TPA. The TPA should define the action to be taken by the sender if an acknowledgment is not received on time or is negative, and should define the imputed significance of acknowledgment. Note that acknowledgments are NOT acknowledged.

Acknowledgment can be used to support non-repudiation. For example, consider the vendor who asserts that a Request For Quotation (RFQ) was not received. If the TPA calls for a positive acknowledgment, the sender will have a record of the acknowledgment message from the recipient. Acknowledgment from a VAN specifying delivery to the recipient also provides evidence to refute repudiation. Assuming good system design, the sender can show how the RFQ system matched each incoming acknowledgment against the list of bidders, and how the sender followed-up promptly when acknowledgments were not received on time.

Similarly, imagine that a bidder attempts to disavow a low bid when an order is received. If the agency issuing the purchase order acknowledges all bids received before "opening" the bids, it can then show that the low bidder did not question the acknowledgment of the receipt of that bid by the agency.

Acknowledgment also supports prompt detection of data corruption and lost messages. Either the EDI systems or the network may fail in such a way that a message is lost in transit and does not reach the recipient's application. Likewise, hardware or software failures may corrupt a message or make it invalid. Because routine human oversight has been eliminated, it is important to be able to detect such failures automatically, and trigger prompt human intervention.

There are five kinds of acknowledgments. Each one is separately described below and shown graphically in Figure 2, p. 39. Not all the acknowledgment types may be necessary for every sender's application; only the most appropriate ones should be used. The detailed implementation of acknowledgments should be based on a risk analysis of the transactions. For example, if the loss resulting from a lost or delayed message can be significant, the time allowed for receipt of an acknowledgment should be relatively short. Similarly, the greater the loss that would result from repudiation, the more extensive the use of acknowledgments should be. If errors in message content could trigger large losses, the recipient application acknowledgment should include validation information.

3.2.1 Sender's EDI System to Sender's Application

The EDI system should tabulate transactions received from the application since last acknowledgment (ack. #1, Fig. 2), recording for each transaction:

(1) the time it was received from the application,

(2) the number of bytes received from the application, and
(3) the status of the transaction.

The status of the transaction should be recorded as one of the following:

(1) queued for translation,
(2) translated error-free,
(3) failed translation and rejected,
(4) passed to the network,
(5) passed to recipient's mailbox (for systems using a VAN),
(6) downloaded by recipient (for systems using a VAN),
(7) acknowledgment received from recipient, or
(8) acknowledgment from recipient overdue.

At regular intervals, the EDI system should send a copy of the tabulation back to the application, which then reconciles the tabulation with its own records to ensure that all transactions were processed and dispatched to the network.

Each application should create and maintain a table of transactions that it passes to or receives from the EDI system. Each table entry should contain enough information to ensure that incorrect operation of the EDI system involving lost or mishandled transactions can be detected. The applications should be able to detect the failure of the EDI system to process outbound transactions in a timely manner.

3.2.2 Network to Sender's EDI System

VANs may provide senders with acknowledgments of receipt of EDI messages by their own and recipients' mailboxes (acks. #2A, #2B, Fig. 2). These reports provide audit trail information about the movement of messages and, as such, they provide evidence of transmission and receipt. This may be particularly important in a dispute caused by an attempt at repudiation. Note that the recipient's mailbox receipt report (#2B) returns through the network.

3.2.3 Recipient's EDI System to Sender's EDI System

Typically, transaction set 997 Functional Acknowledgment is generated automatically by the recipient's EDI system when a valid transaction set is received. The functional acknowledgment simply acknowledges receipt of the message, but it is not an operational "acceptance" of the intent of the transaction set. The TPA should be clear as to the meaning of a 997 with respect to each transaction set defined in the TPA. For example, it should not be taken to mean "acceptance" of a purchase order. Note that this acknowledgment (ack. #3, Fig. 2) also flows back through the network.

Functional acknowledgments should be assured to be generated by the EDI system in a timely manner. The TPA may call for a trading partner to send functional acknowledgments for specific transaction sets within a specified time after receipt. Failure to acknowledge promptly will trigger an "acknowledgment not received" action by the sender, and require wasteful corrective actions by both partners. The system design should provide for the situation in which the sender initiates an "acknowledgment not received" action but later receives a positive acknowledgment from the recipient.

3.2.4 Recipient's Application to Recipient's EDI System

The recipient's EDI system can maintain a log of incoming transaction sets that it has passed to the applications. Periodically (e.g., daily), each application can acknowledge to the EDI system the number and types of transaction sets received and processed (ack. #4, Fig. 2). This will provide data necessary for the EDI system to detect lost transaction sets.

3.2.5 Recipient's Application to Sender's Application

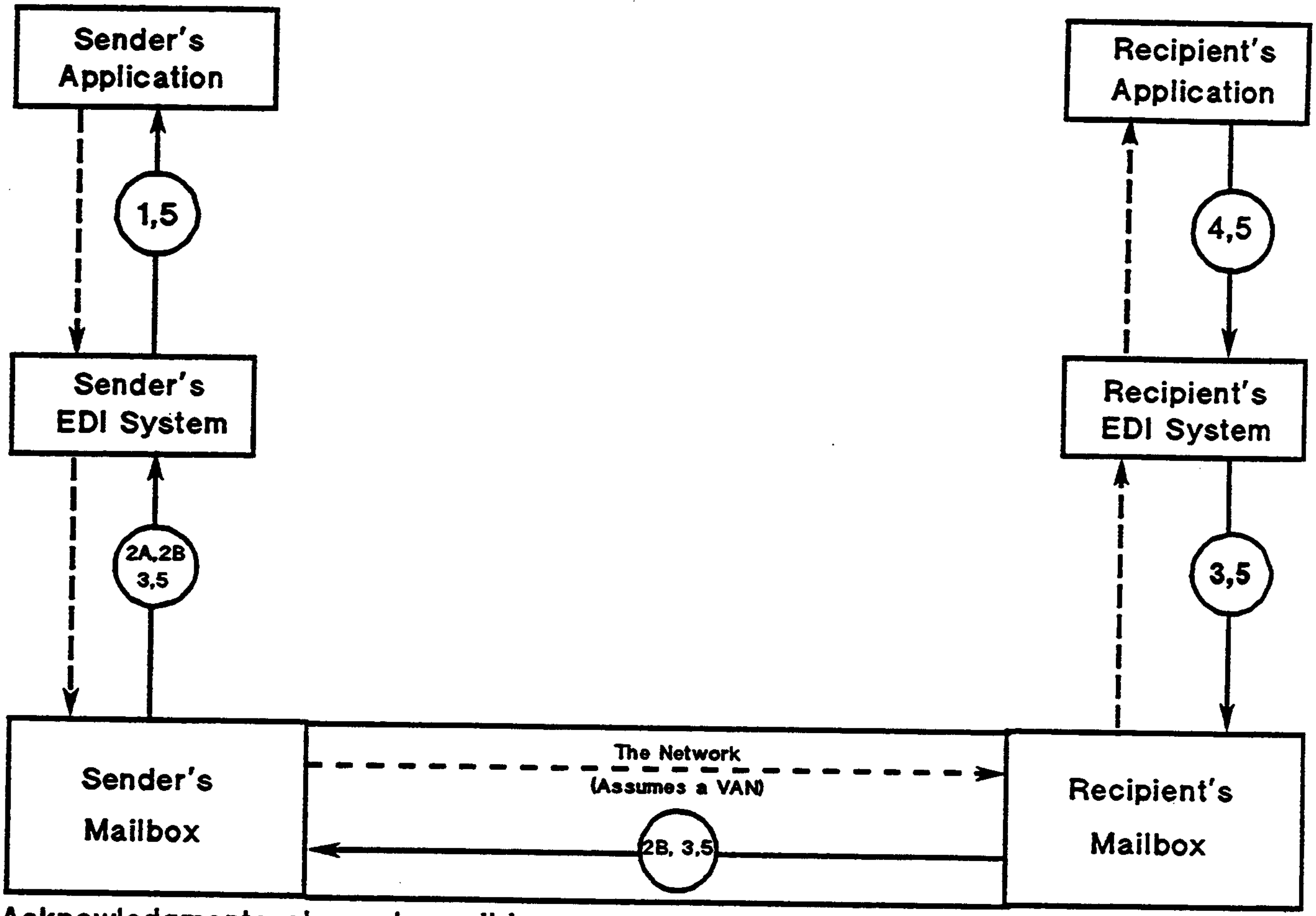
As specified by the TPA, the recipient's application that receives the translated transaction set acknowledges receipt, and indicates the action that the recipient is going to take. This acknowledgment is an action acknowledgment (ack. #5, Fig. 2), as distinguished from the 997 Functional Acknowledgment. An action acknowledgment transaction set might be, for example, an 824 Application Advice, 855 Purchase Order Acknowledgment, or 856 Ship Notice/Manifest. It is this acknowledgment that signals "acceptance or rejection" of the sender's transaction. Note that this acknowledgment also flows back through the network.

If an electronic document, passed to the recipient as an EDI transaction set, has been signed by an individual, the acknowledgment should include the imputed identity of the signer. The TPA should specify a time limit within which the sender, after receiving an acknowledgment, must question the identity if it is wrong. The acknowledgment may include information that the sender's application can use to verify that the information in the message was received intact without modification or corruption. For example, the acknowledgment might include hash totals of part numbers and monetary amounts, or it might indicate that a Message Authentication Code was confirmed.

3.3 Techniques for Applications

3.3.1 Sequential Numbering of Sender's Transactions for Each Recipient

Each application that generates sender transactions should assign an identifying number to each transaction, and include the number in the transaction set sent to the recipient. Transactions sent to a particular recipient should be sequentially numbered.



Acknowledgments shown by solid arrows
Message paths shown by dashed arrows

Acknowledgment Types

1: Status of sender's transactions

2A: Receipt by sender's mailbox

2B: Receipt by recipient's mailbox

3: Functional acknowledgment

4: Status of recipient's transactions

5: Action acknowledgment

Figure 2. Typical EC System Acknowledgments.

<u>Case Study</u>: A buying partner sent an EDI purchase order for 500 aluminum ladders to a selling partner. Because of a badly worded transmission error report, the buyer mistakenly concluded that the transaction set had not been received, and sent it again. Because the purchase order did not include a unique number, the seller could not detect the duplication. As a result, the seller fabricated and shipped 1,000 ladders to the buyer.

It is essential to include a unique sequence number in each outgoing operational transaction to ensure that the recipient can detect duplicate messages. Note, however, that the recipient cannot detect missing or out-of-sequence transactions unless the messages include sequence numbers that are unique to each recipient. Information messages, such as an RFQ (Request For Quotation) or price list update, probably do not require sequence numbers since the content of the message, e.g., an internal "publication" date, typically discloses duplicates. In other words, no harm is done if the recipient receives two copies of the same RFQ. Acknowledgments of sequence-numbered transaction sets should include the sequence number of the transaction set being acknowledged, so they do not require their own sequence numbers.

3.3.2 Testing For and Reporting of Duplicate Messages

Recipient applications should test incoming messages to detect duplicate messages, and report them to the sender.

The TPA should define the requirement for a recipient to detect duplicate messages and transaction sets, and the action that the recipient is to take when a duplicate is detected. A minimum default condition could be to ignore duplicate message. However, since a duplicate message is a symptom of an operating error or a system failure, it is good security practice to report the duplication to the sender, and for the sender to diagnose and correct the cause.

3.3.3 Error Handling

Applications should be enhanced to resolve error conditions, automatically if possible, or by generating exception reports for human resolution. It is important to ensure that error handling is complete and correct. The sender application must be able to detect and resolve correctly (a) failures to transmit messages, (b) failures to receive acknowledgments in a timely manner, and (c) acknowledgments that indicate that alterations to messages have occurred.

3.3.4 Testing For Invalid and Suspect Transactions

Recipient applications should perform traditional edit checks of incoming transactions, and should also verify the "reasonableness" of transactions.

There may have been significant reasonableness checking by human operators in the paper-driven system; this human oversight may not be completely documented. It is essential to identify all human oversight during the EC design and implementation phase, and to decide how that oversight is to be replaced with automated processing. For example, consider a recipient application that processes purchase orders from many other trading partners. The application might be modified to construct a profile of typical purchase orders for each of the other trading partners. As each purchase order is received, it could be compared with the sender's profile. If the purchase order falls outside the limits defined by the profile, it could diverted for review by an experienced staff member or to a computerized "expert system" for further analysis.

3.3.5 Assurance of Message Integrity

Both parties to a data interchange want reasonable assurance that the critical information included in a message when composed is unchanged when received. The concern for potential loss requires that, if an action is to be taken as the result of a message, the action is taken on the basis of correct data.

1) Use of Hash Totals

One common and elementary technique that helps assure message integrity is the inclusion of "hash totals" in the message. A hash total is a summation for checking purposes of similar fields in a file, such as fields containing part numbers, that would otherwise not be summed. This concept has been adopted for EDI. For example, the X12 850 Purchase Order transaction set allows the sender to include the sum of the value of the quantities added, as well as the total transaction amount. The TPA should require that hash totals be provided by the sender and verified by the recipient. Figure 3 illustrates the concept. This security measure is quite simple to implement.

Purchase Order No. 123-456

Quantity	Part Number	Unit Price	Total
3	1234	$ 123.45	$ 370.35
5	6678	$ 22.44	$ 112.20
----	-------	---------	---------
8	7912*	145.89*	$ 482.55

*: Hash totals with no real-world meaning.

Figure 3. An Example of a Purchase Order With Hash Totals.

Verification of hash totals could be combined with reasonableness checking, as discussed in Section 3.3.4 above.

2) Secure Hash Standard

Hash totals only protect specific data fields in the transaction sets. It is also possible to protect an entire transaction set against undetected alteration or corruption. One way to do this is to use the Secure Hash Algorithm (SHA), specified in recently adopted FIPS PUB 180. The SHA accepts, as input, a message of any length in bits less than 2 to the 64th power, and generates a 160-bit output called a message digest. The SHA is called secure because it is not feasible to find a way to alter a message without altering the message digest. Thus, if a message is altered, the message digest calculated by the recipient will not match the digest attached to the message by the sender. FIPS PUB 180 includes a complete description of the SHA.

It is extremely unlikely that the body of a message and its message digest could both be corrupted accidentally such that the corrupted digest matches the corrupted message. Therefore the SHA will protect a transaction set against accidental alteration, but not against deliberate alteration. An intruder could deliberately modify a message, then calculate a new message digest and substitute it for the original digest. Thus, the message would appear unmodified to the recipient. If there is a significant risk of deliberate modification of a transaction set, then a more secure form of message authentication may be appropriate.

3.3.6 Digital Signature Algorithm

A digital signature provides additional security. It enables a message recipient to verify the originator of the message as well as the message content.

A Digital Signature Algorithm (DSA) which uses the SHA is currently being considered for adoption as a FIPS PUB. The DSA employs two cryptographic keys for each user. Each user has a public key that is known by all trading partners, and a private key that is kept secret. The message to be sent serves as input to the SHA; the output of the SHA operation is the message digest. The message digest and the sender's private key are used in a signing algorithm to calculate the digital signature. The recipient receives both the message and the digital signature.

A signature verification algorithm is used by the recipient to authenticate the signer. This algorithm uses, as inputs, the sender's public key, the received digital signature, and the message digest recalculated with the SHA from the received message. The verification algorithm recalculates one of two signature components. If the recalculated component matches the component as received, the signer is authenticated and the received message is identical to that sent. If the signature fails to verify, the recipient must ask for the message to be retransmitted. The process is shown graphically in Figure 4, p. 44.

This public key technique has the advantage that it can be used in more than one trading partnership. Each user's key pair may be used for message interchange with any trading partner, and the private key need never be exchanged or revealed. However, for general implementation, a high-security administrative system needs to be in place. This system would provide secure distribution of private keys, and a trustworthy source of public key information. As of this writing, no such general system is available.

Non-cryptographic Originator Authentication: A simpler but less assured system for originator authentication is as follows. For each trading partner pair, the recipient generates unique lists of random numbers, and sends one list to each signatory in the sender's organization. The means of delivery used must protect the lists against compromise. Each time an individual wants to sign a message, that individual simply adds the next number on his or her unique list to the message, and then crosses off the number, making a note of the time and date it was used. The recipient verifies that the signature number on each message is the next number on the signatory's list, and the recipient includes the number on the message acknowledgment. If someone else in the sender's organization or an outsider gets access to the list and uses the next number, the acknowledgment will alert the authorized individual. This method, unlike the DSA, does not provide an integrity check for the whole message.

3.3.7 Message Confidentiality

If the risk analysis of a planned EC system shows that there is a significant possibility that sensitive messages will be disclosed while being communicated, and that the disclosure would be seriously detrimental, the messages should be encrypted. The cost to encrypt will include (a) purchase, operation and maintenance of cryptographic devices, (b) the cost to manage and distribute the cryptographic keys, and (c) the cost of any additional network data transmission capacity required (encryption usually increases the number of bytes in a message). The costs of protection and the potential losses due to disclosure could be factored into a QRA.

3.3.8 Audit Trails of Transaction Processing

To support non-repudiation, and facilitate recovery from errors and breakdowns, each application should maintain an audit trail of the processing of transactions.

If there is a significant risk of repudiation, the sender's application should maintain an adequate audit trail of the transactions that the application initiates. The audit trail should make it

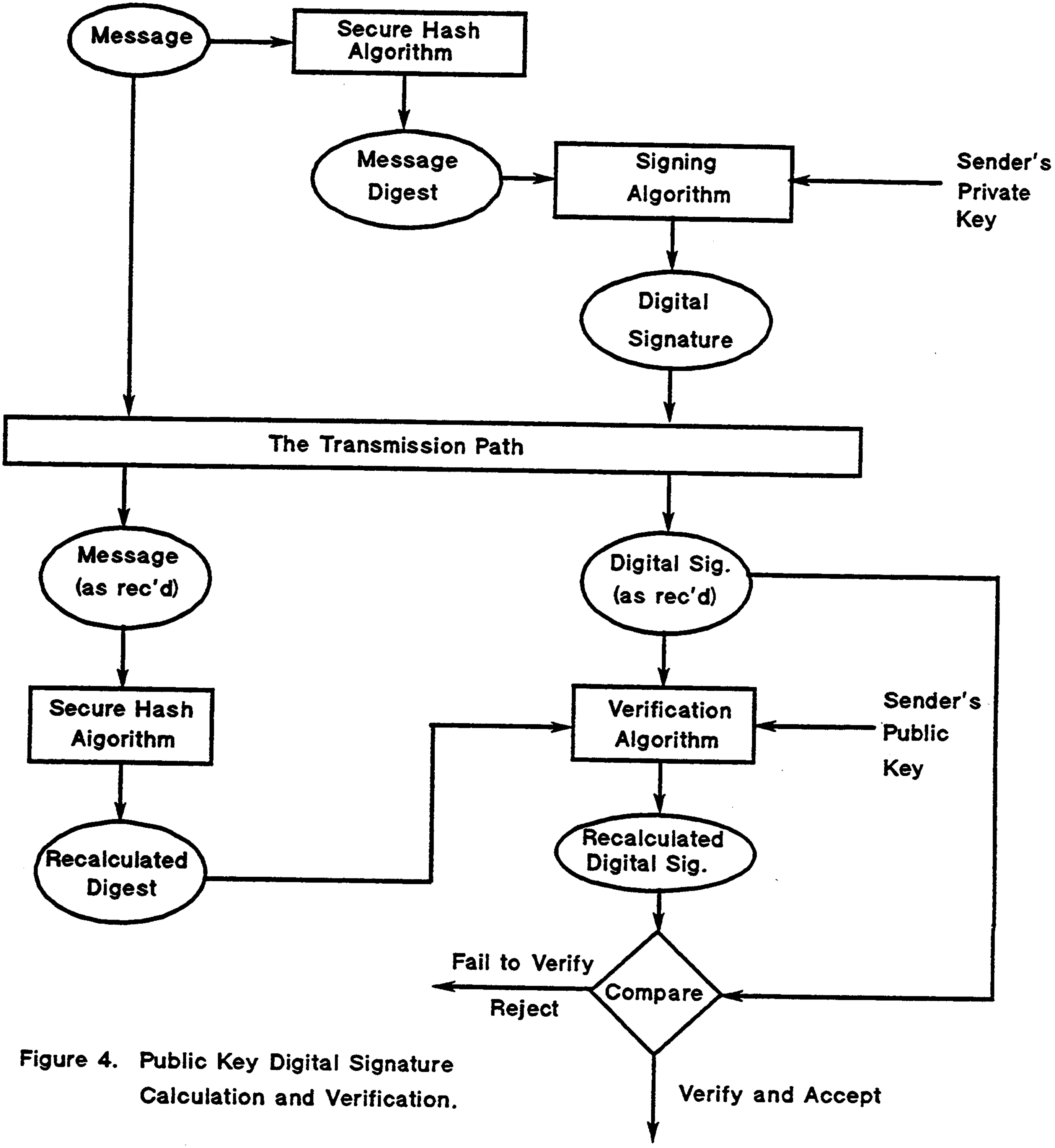


Figure 4. Public Key Digital Signature Calculation and Verification.